Sistemi Operativi
7 luglio 2011
Compito B

Si risponda ai seguenti quesiti, giustificando le risposte.

1. (a) Si indichino i parametri in base ai quali viene valutato un algoritmo di scheduling della CPU.
   (b) Quali dovrebbero essere le caratteristiche di un algoritmo di scheduling per un sistema real-time?

   Risposta:

   (a) (3 punti) I parametri in base ai quali viene valutato un algoritmo di scheduling della CPU sono i seguenti:
   - *utilizzo della CPU:* mantenere la CPU più carica possibile;
   - *throughput:* numero di processi completati nell'unità di tempo;
   - *tempo di turnaround:* tempo totale impiegato per l'esecuzione di un processo;
   - *tempo di attesa:* quanto tempo un processo ha atteso in coda ready;
   - *tempo di risposta:* quanto tempo si impiega da quando una richiesta viene inviata a quando si ottiene la prima risposta (non l'output);
   - *varianza del tempo di risposta:* quanto il tempo di risposta è variabile.

   (b) (2 punti) Per un sistema real-time le caratteristiche di un algoritmo di scheduling dovrebbero garantire il rispetto delle deadline per evitare la perdita di dati e la prevedibilità (ad esempio per evitare il degrado della qualità nei sistemi multimediali).

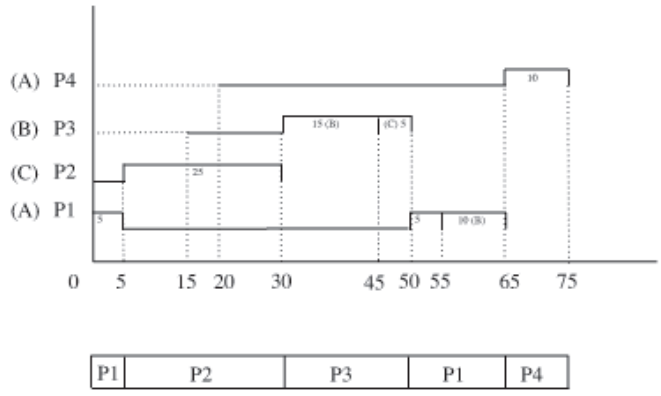
2. Si consideri un sistema con scheduling a priorità con tre code, A, B, C, di priorità crescente, con prelazione tra code. Le code A e B sono round robin con quanto di 10 e 15 ms, rispettivamente; la coda C è FCFS. Se un processo nella coda A o B consuma il suo quanto di tempo, viene spostato in fondo alla coda B o C, rispettivamente. Un processo prelazionato torna all'inizio della propria coda.
   (a) Nelle code A, B, C entrano i seguenti processi:

	coda	arrivo	burst
P_1	A	0	20ms
P_2	C	5	25ms
P_3	B	15	20ms
P_4	A	20	10ms

   Si determini:
   1. il diagramma di GANTT relativo all'esecuzione dei quattro processi;
   2. il tempo di reazione medio;
   3. il tempo di turnaround medio.

   Risposta:

   (a) (3 punti)
   1. Il diagramma di Gantt è il seguente:



   2. (2 punti) Tempo di reazione medio = $\frac{0+0+15+45}{4} = \frac{60}{4} = 15$ ms.
   3. (2 punti) Tempo di turnaround medio = $\frac{65+25+35+55}{4} = \frac{180}{4} = 45$ ms.

Sistemi Operativi
7 luglio 2011
Compito B

3. Si consideri un sistema con memoria paginata a un livello, la cui page table sia mantenuta in memoria principale. Il tempo di accesso alla memoria principale sia $t = 40ns$.
   (a) Qual è il tempo effettivo di accesso alla memoria?
   (b) Aggiungendo un TLB, con tempo di accesso $\epsilon = 2ns$, quale hit rate dobbiamo avere per un degrado delle prestazioni del 20% rispetto a t?
   (c) E con una paginazione a due livelli?

   Risposta:

   1. (2 punti) Il tempo effettivo di accesso alla memoria è $2t$, ovvero, 80 ns; infatti sono necessari 40 ns per accedere alla page table e 40 ns per accedere alla locazione nel frame fisico in memoria.
   2. (3 punti) Un degrado del 20% rispetto a t significa un EAT pari a $1,2 \cdot t$, ovvero, 48 ns. Quindi si ha quanto segue (α rappresenta l'hit rate):

$$\begin{array}{rcc} EAT & = & \epsilon + \alpha t + (1-\alpha)2t \\ 48 & = & 2 + 40\alpha + (1-\alpha) \cdot 80 \\ 48 & = & 82 - 40\alpha \end{array}$$

      da cui si ricava $\alpha = \frac{34}{40} = 0,85$ (85%).
   3. (3 punti) Con una paginazione a due livelli si ha quanto segue:

$$\begin{array}{rcc} EAT & = & \epsilon + \alpha t + (1-\alpha)3t \\ 48 & = & 2 + 40\alpha + (1-\alpha) \cdot 120 \\ 48 & = & 122 - 80\alpha \end{array}$$

      da cui si ricava $\alpha = \frac{74}{80} = 0,925$ (92,5%).

4. Si consideri un processo che generi la seguente stringa di riferimenti alle pagine virtuali:

$$0\ 2\ 1\ 0\ 4\ 0\ 2\ 1\ 1\ 0\ 5\ 3\ 2$$

   (a) Se il processo ha 3 frame, gestiti LRU, quanti page fault vengono generati?
   (b) Qual è il numero minimo di frame necessario per minimizzare i page fault?

   Risposta:

   (a) (3 punti) Simuliamo il funzionamento di LRU nel caso della reference string data:

	0	2	1	0	4	0	2	1	1	0	5	3	2
		0	2	1	0	4	0	2	2	1	0	5	3
			0	2	1	1	4	0	0	2	1	0	5
					2	2	1	4	4	4	2	1	0
											4	2	1
												4	4
	P	P	P		P		P	P			P	P	P

   Si verificano quindi nove page fault.

   (b) (2 punti) Il minimo numero di page fault è 6 page fault (perché il processo accede a 6 pagine). Per determinare il numero minimo di frame per avere solo 6 page fault, si può sfruttare la *distance string*, che nel caso in questione risulta essere la seguente:

$$\infty\ \infty\ \infty\ 3\ \infty\ 2\ 4\ 4\ 1\ 3\ \infty\ \infty\ 5$$

   Si ricorda che la distance string rappresenta la distanza fra la posizione di una pagina nel modello e la prima posizione, ovvero, quella nella prima riga della matrice (contando anche la casella di partenza) nel momento in cui la pagina stessa viene riferita. Se una pagina non è presente nella matrice, allora la sua distanza, quando viene riferita è ∞.

   Indichiamo ora con C_i il numero di volte che il numero i compare nella distance string; nel caso in questione abbiamo: $C_1 = 1$, $C_2 = 1$, $C_3 = 2$, $C_4 = 2$, $C_5 = 1$, $C_\infty = 6$. Indicando poi con m il numero di frame e con n il numero più grande che compare nella distance string,

indichiamo con $F_m = \Sigma_{k=m+1}^{n} C_k + C_\infty$ il numero di page fault che si verificano con m frame e con la reference string data. L'intuizione è la seguente: se ho a disposizione m frame i page fault saranno provocati dai riferimenti a pagine che "distano" almeno $m+1$ dal top della matrice e dal numero di ∞ (ovvero da riferimenti a pagine non ancora presenti nel modello). Nel nostro caso abbiamo: $F_1 = 12$, $F_2 = 11$, $F_3 = 9$, $F_4 = 7$, $F_5 = 6$; quindi il numero minimo di frame che minimizza i page fault è 5.

5. Si spieghi come funziona l'allocazione concatenata dei blocchi di un disco. Che problemi possono verificarsi?

   Risposta: (3 punti) Lo schema di allocazione concatenata prevede che ogni blocco dati abbia una parte di esso riservata alla memorizzazione dell'indirizzo del blocco successivo (l'ultimo blocco viene marcato con un indirizzo non valido, e.g., -1). In questo modo l'implementazione di un file consiste in una lista di blocchi concatenati ed è sufficiente memorizzare per ogni file nella directory entry il blocco iniziale e quello finale. I pregi consistono essenzialmente nella semplicità dell'implementazione, mentre gli aspetti negativi sono la fragilità dello schema (se si corrompe un puntatore della catena, si perde tutta la parte del file da quel punto in poi) e la mancanza di supporto dell'accesso diretto (seek); infatti per accedere al blocco i-esimo bisogna prima scorrere la catena di puntatori dei precedenti i-1 blocchi.

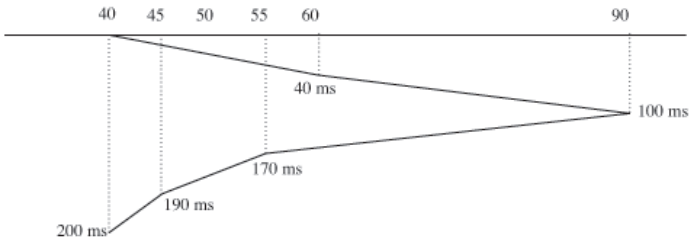
   Questo schema non soffre di frammentazione esterna, dato che i blocchi assegnati ad un file possono essere sparsi ovunque nel disco (non devono necessariamente essere contigui); infatti, per aggiungere un blocco ad un file è sufficiente aggiornare il puntatore dell'ultimo blocco in modo che punti ad uno dei blocchi liberi su disco (marcando l'indirizzo di quest'ultimo come indirizzo finale, e.g., -1 ed aggiornando la directory entry corrispondente).

6. Si consideri un disco gestito con politica LOOK. Inizialmente la testina è posizionata sul cilindro 40; lo spostamento ad una traccia adiacente richiede 2 ms. Al driver di tale disco arrivano richieste per i cilindri 90, 45, 40, 60, 55, rispettivamente agli istanti 0 ms, 20 ms, 30 ms, 40 ms, 80 ms. Si trascuri il tempo di latenza.

   1. In quale ordine vengono servite le richieste?
   2. Il tempo di attesa di una richiesta è il tempo che intercorre dal momento in cui è sottoposta al driver a quando viene effettivamente servita. Qual è il tempo di attesa medio per le cinque richieste in oggetto?

   Risposta:

   1. (3 punti) Le richieste vengono servite nell'ordine 60, 90, 55, 45, 40:



   2. (1 punto) Il tempo di attesa medio per le cinque richieste in oggetto è $\frac{(100-0)+(190-20)+(200-30)+(40-40)+(170-80)}{5} = \frac{100+170+170+0+90}{5} = \frac{530}{5} = 106\ ms.$